Iterated Transduction on Unary Languages⋆ (short paper)

Martin Kutrib[1], Andreas Malcher[1], Carlo Mereghetti[2], and Beatrice Palano[3]

[1] Institut für Informatik, Universität Giessen, Arndtstr. 2, 35392 Giessen, Germany
{kutrib,andreas.malcher}@informatik.uni-giessen.de
[2] Dipartimento di Fisica "Aldo Pontremoli", Università degli Studi di Milano
via Celoria 16, 20133 Milano, Italy, carlo.mereghetti@unimi.it
[3] Dipartimento di Informatica "G. degli Antoni", Università degli Studi di Milano
via Celoria 18, 20133 Milano, Italy, palano@unimi.it

Abstract. An *iterated uniform finite-state transducer* executes the same length-preserving transduction in iterative sweeps. The first sweep occurs on the input string, while any subsequent sweep processes the output of the previous one. We focus on *unary inputs*. We show that any unary regular language can be accepted by a deterministic iterated uniform finite-state transducer with at most $\max\{2\varrho, p\}+1$ states, ϱ and p being the greatest primes in the factorization of the, respectively, pre-periodic and periodic part of the language. This state cost cannot be improved by using nondeterminism, and is definitely lower than the state costs of equivalent classical models of finite-state automata.

Keywords: Iterated transducers; Unary languages; State complexity

1 Introduction

The notion of an iterated uniform finite-state transducer (IUFST) has been introduced in [20, 21, 25]. It consists of a length-preserving finite-state transducer that works in iterative sweeps from left to right on its input tape. In the first sweep the input string is processed, while any further sweep operates on the output of the previous sweep. The model is uniform in that every sweep always starts from the same initial state on the leftmost tape symbol, and operates the same transduction rules at each computation step. An input string is accepted whenever the transducer halts in an accepting state at the end of a sweep.

A theoretical investigation of IUFSTs is motivated by the fact that iterated or cascade transductions show up in numerous fields of computer science, e.g., in natural language processing [17], in compiler design [1], in the celebrated Krohn-Rhodes decomposition theorem [18]. Again, cascades of deterministic pushdown transducers as language accepting devices have been studied in [15]. In [11, 26], iterated finite-state transducers as language generating devices have been settled.

⋆ Extended version presented at the *47th Int. Conf. SOFSEM*, January 25–29, 2021, Bolzano-Bozen, Italy, [24].

Notice that only these latter contributions introduce a notion of "uniformity" on iterated transduction, in that always the same transducer is iteratively applied.

Deterministic and nondeterministic IUFSTs (the nondeterministic model denoted by NIUFST) have been deeply studyied in [20, 21, 23, 25]. For a constant number of sweeps, IUFSTs and NIUFSTs characterize the class of regular languages. For a non-constant number of sweeps, non-regular languages can be accepted as long as an at least logarithmic number of sweeps is provided, and infinite proper language hierarchies depending on sweep complexity are shown. Recently, in [22], IUFSTs and NIUFSTs have been enhanced with the possibility of two-way motion, implying a sweep alternation from left to right and from right to left. Indeed, IUFSTs and NIUFSTs have been investigated within the realm of *descriptional complexity*, where device *size* is under consideration (see, e.g., [2–10, 19]). In descriptional complexity, techniques from several areas are employed [12, 13, 28], yielding results of both theoretical and practical interest [16].

In this paper, we construct size-efficient iterated transducers for general *unary regular languages*. We give our construction step by step, starting from finite unary languages. We show that finite unary languages with words of length up to ℓ can be accepted by an IUFST with 2ϱ states, ϱ being the greatest prime in the factorization of ℓ. Next, we switch to unary periodic languages, and prove that any n-periodic unary language can be accepted by an IUFST with p states, p being the greatest prime in the factorization of n. By combining these two results, we get that any unary regular language is accepted by an IUFST with at most $\max\{2\varrho, p\} + 1$ states. We then show that these state costs cannot be improved by using nondeterminism, and that they are definitely lower than the state costs of equivalent classical models of finite-state automata.

Due to the page limit, proofs and further results (some of which may be found in [24]) have been omitted.

2 Definitions and Preliminaries

By the Fundamental Theorem of Arithmetic, any integer $n > 1$ univocally factorizes as $n = \prod_{i=1}^{s} p_i^{\alpha_i}$, for primes $p_1 < \cdots < p_s$ and integers $\alpha_i > 0$. For any $n > 1$, we let $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ be the ring of the remainders modulo n. Let Σ^* be the set of words on a finite alphabet Σ, including the empty word. The length of a word w is denoted by $|w|$. A language on Σ is any set $L \subseteq \Sigma^*$.

A *nondeterministic iterated uniform finite-state transducer* (NIUFST) is a 8-tuple $T = \langle Q, \Sigma, \Delta, q_0, \triangleright, \triangleleft, \delta, F \rangle$, with Q the set of *internal states*, Σ the set of *input symbols*, Δ the set of *output symbols*, $q_0 \in Q$ the initial state, $\triangleright \in \Delta \setminus \Sigma$ and $\triangleleft \in \Delta \setminus \Sigma$ the *left and right endmarkers*, respectively, $F \subseteq Q$ the set of *accepting states*, and $\delta\colon Q \times (\Sigma \cup \Delta) \to 2^{Q \times \Delta}$ the partial *transition function*. The NIUFST T *halts* if the transition function is undefined or T enters an accepting state at the end of a sweep. The transduction is applied in multiple sweeps, and in any but the initial sweep it processes an output of the previous sweep. So, the transition function depends on symbols in $\Sigma \cup \Delta$. Let $T(w)$ be the set of possible outputs yielded by T in a complete sweep on the input string $w \in (\Sigma \cup \Delta)^*$.

In a computation on input $w \in \Sigma^*$, the NIUFST T yields a sequence of words $w_1, \ldots, w_i, w_{i+1}, \ldots \in (\Sigma \cup \Delta)^*$, with $w_1 \in T(\triangleright w \triangleleft)$ and $w_{i+1} \in T(w_i)$ for $i \geq 1$. An iterated uniform finite-state transducer is *deterministic* (IUFST) whenever $|\delta(p, x)| \leq 1$, for all $p \in Q$ and $x \in (\Sigma \cup \Delta)$. In this case, we simply write $\delta(p, x) = (q, y)$ instead of $\delta(p, x) = \{(q, y)\}$ assuming that the transition function is a mapping $\delta\colon Q \times (\Sigma \cup \Delta) \to Q \times \Delta$.

A computation is halting if there exists an $r \geq 1$ such that T halts on w_r, thus performing r sweeps. The input word $w \in \Sigma^*$ is *accepted* by T if at least one computation on w halts at the end of a sweep in an accepting state. Otherwise it is *rejected*. Indeed, the output of the last sweep is not used. The language accepted by T is the set $L(T) \subseteq \Sigma^*$ defined as $L(T) = \{\, w \in \Sigma^* \mid w \text{ is accepted by } T \,\}$.

For nondeterministic computations and some complexity bound, several language acceptance modes are usually considered in the literature. Here, we deal with the number of sweeps as complexity measure, and adopt the so-called *accept mode* of acceptance [27]. A language is accepted in the *accept mode* if all accepting computations obey the complexity bound. More precisely, given a function $s\colon \mathbb{N} \to \mathbb{N}$, an iterated uniform finite-state transducer T has *sweep complexity* $s(n)$ if for all $w \in L(T)$ all accepting computations on w halt after at most $s(|w|)$ sweeps. In this case, we add the prefix $s(n)$- to the notation of the device. It is easy to see that 1-IUFSTs (resp., 1-NIUFSTs) are deterministic (resp., nondeterministic) finite-state automata (DFAs and NFAs, respectively).

A language $L \subseteq \Sigma^*$ is *unary* whenever built on a single-letter alphabet. In this case, we let $L \subseteq 0^*$. A unary language $L \subseteq 0^*$ is n-periodic if there exists $\mathcal{R} \subseteq \mathbb{Z}_n$ such that $L = \{\, 0^{c \cdot n + R} \mid c \geq 0 \text{ and } R \in \mathcal{R} \,\}$. We will always be assuming that n is the minimal value defining L. This is usually referred to as L being *properly* n-periodic. To emphasize periodicity and $\mathcal{R} \subseteq \mathbb{Z}_n$, we express L in the form $L_{n,\mathcal{R}}$. By pumping arguments, we get that n states are necessary and sufficient for DFAs and NFAs to accept $L_{n,\mathcal{R}}$. The minimal DFA for $L_{n,\mathcal{R}}$ consists of a single cordless cycle of n states, with an initial state and final states settled according to $\mathcal{R}$. On the other hand, for n factorizing as $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \cdots \cdot p_s^{\alpha_s}$, we have that any two-way DFA and NFA, or isolated cut-pint probabilistic finite automaton (PFA) for $L_{n,\mathcal{R}}$ must have at least $\sum_{i=1}^{s} p_i^{\alpha_i}$ states [29, 30].

Any unary regular language is the disjoint union of a finite language and an ultimately periodic language. So, it is defined by three parameters ℓ, n, $\mathcal{R} \subseteq \mathbb{Z}_n$:

$$L_{\ell,n,\mathcal{R}} = L_\ell \cup \{\, 0^{\ell + c \cdot n + R} \mid c \geq 0 \text{ and } R \in \mathcal{R} \,\},$$

The set L_ℓ, called the pre-periodic part of $L_{\ell,n,\mathcal{R}}$, is a finite unary language with strings of length not exceeding ℓ. The set $\{\, 0^{\ell + c \cdot n + R} \mid c \geq 0 \text{ and } R \in \mathcal{R} \,\}$ is the periodic part of $L_{\ell,n,\mathcal{R}}$. As usual, we assume that the parameters ℓ and n are the smallest possible defining $L_{\ell,n,\mathcal{R}}$. Note that $L_{\ell,n,\mathcal{R}}$ is accepted by a DFA whose state digraph features an initial path of ℓ states joined to a cordless cycle of n states. Clearly, the above recalled state lower bound for two-way DFAs and NFAs, or isolated cut-point PFAs for $L_{n,\mathcal{R}}$ carries over to $L_{\ell,n,\mathcal{R}}$ as well. By simulation results in [14, 31], if $L_{\ell,n,\mathcal{R}}$ is accepted by a b-state NFA or two-way NFA, then we can assume $\ell = O(b^2)$ and $n = \mathrm{e}^{\Theta(\sqrt{b \cdot \log b})}$.

3 Iterated Transduction and Unary Regular Languages

We present our construction of state-efficient IUFSTs for unary regular languages by first focusing on unary *periodic* languages, then considering *finite* unary languages, and finally getting to *general* unary regular languages.

3.1 Unary Periodic Languages

For reader's ease of mind, we start by considering unary periodic languages of the form $L_n = \{\, 0^{c \cdot n} \mid c \geq 0 \,\}$, for $n \geq 1$.

Theorem 1. *Let $n \geq 2$ factorize as $n = \prod_{i=1}^{s} p_i^{\alpha_i}$, with $\alpha_i > 0$. The language L_n can be accepted by a p_s-state r-IUFST with $r = \sum_{i=1}^{s} \alpha_i$ sweeps.*

The minimality – in terms of number of states – of the IUFST for L_n designed in Theorem 1 is provided by a pumping argument in the following theorem, which also shows that nondeterminism cannot help in reducing state size:

Theorem 2. *Let $n \geq 2$ factorize as $n = \prod_{i=1}^{s} p_i^{\alpha_i}$, with $\alpha_i > 0$. Any IUFST and NIUFST accepting L_n must use at least p_s states, regardless of the number of performed sweeps.*

Let us now move on to a slightly different version of L_n. Precisely, for any $n \geq 1$ and a fixed remainder $R \in \mathbb{Z}_n \setminus \{0\}$, we let $L_{n,R} = \{\, 0^{c \cdot n + R} \mid c \geq 0 \,\}$. The next theorem shows that this modification of L_n does not increase state and sweep complexity of acceptance on iterated transducers.

Theorem 3. *Let $n \geq 2$ factorize as $n = \prod_{i=1}^{s} p_i^{\alpha_i}$, with $\alpha_i > 0$. The language $L_{n,R}$ can be accepted by a p_s-state r-IUFST with $r = \sum_{i=1}^{s} \alpha_i$ sweeps.*

Finally, we come to tackle the acceptance of a general unary n-periodic language $L_{n,\mathcal{R}}$, for a fixed set $\mathcal{R} \subseteq \mathbb{Z}_n$: $L_{n,\mathcal{R}} = \{\, 0^{c \cdot n + R} \mid c \geq 0 \text{ and } R \in \mathcal{R} \,\}$.

Theorem 4. *Let $n \geq 2$ factorize as $n = \prod_{i=1}^{s} p_i^{\alpha_i}$, with $\alpha_i > 0$. The language $L_{n,\mathcal{R}}$ can be accepted by a p_s-state r-IUFST with $r = \sum_{i=1}^{s} \alpha_i$ sweeps.*

We notice that from Theorem 2 one may obtain that any IUFST and NIUFST accepting $L_{n,\mathcal{R}}$ must use at least p_s states, regardless the number of performed sweeps. In addition, as recalled in Section 2, we remark that n states are necessary and sufficient for DFAs and NFAs to accept $L_{n,\mathcal{R}}$, while $\sum_{i=1}^{s} p_i^{\alpha^i}$ states are necessary for two-way DFAs and NFAs, and one-way isolated cut point PFAs.

3.2 Unary Finite Languages

For any positive integer ℓ, let L_ℓ be any unary language whose longest word has length ℓ. In what follows, we assume $\ell \geq 2$. The case $\ell = 1$ can be trivially managed by a 2-state DFA seen as a transducer.

Theorem 5. *Let $\ell \geq 2$ factorize as $\ell = \prod_{i=1}^{r} \varrho_i^{\beta_i}$, with $\beta_i > 0$. The language L_ℓ can be accepted by a $(2\varrho_r)$-state t-IUFST with $t = \sum_{i=1}^{r} \beta_i$ sweeps.*

Any classical finite-state automata for L_ℓ needs at least ℓ states. By a pumping argument as in Theorem 2, iterated transduction needs at least ϱ_r states.

3.3 General Unary Regular Languages

Finally, let us put things together. As addressed in Section 2, a general (infinite) unary regular language consists of the disjoint union of a (possibly empty) finite pre-periodic language and a ultimately periodic language. Hence, it can be defined by three parameters ℓ, n, and $\mathcal{R} \subseteq \mathbb{Z}_n$ as

$$L_{\ell,n,\mathcal{R}} = L_\ell \cup \{\, 0^{\ell+c\cdot n+R} \;\mid\; c \geq 0 \text{ and } R \in \mathcal{R}\,\},$$

where, with a slight abuse with respect to notation in Section 3.2, we intend L_ℓ as a finite unary language that is accepted by an ℓ-state DFA. So, differently from Section 3.2, from now on L_ℓ does not necessarily contain the unary word of length ℓ. In the following theorem, we consider $\ell, n \geq 2$. Otherwise, we have languages that can be trivially dealt with by our constructions in this paper.

Theorem 6. *Let $\ell, n \geq 2$ factorize as $n = \prod_{i=1}^{s} p_i^{\alpha_i}$, $\ell = \prod_{i=1}^{r} \varrho_i^{\beta_i}$, $\alpha_i, \beta_i > 0$. The unary language $L_{\ell,n,\mathcal{R}}$ can be accepted by an x-state t-IUFST, where $x = \max\{2\varrho_r, p_s\} + \xi$, $\xi = 0$ if $\varrho_r < p_s$ and 1 otherwise, and $t = \sum_{i=1}^{s} \alpha_i + \sum_{i=1}^{r} \beta_i$.*

By adapting the pumping argument in Theorem 2, we get that not less than $\max\{\varrho_r, p_s\}$ states can be used to accept $L_{\ell,n,\mathcal{R}}$ on iterated transducers. On the other hand, as recalled in Section 2, $\ell + n$ states are necessary and sufficient for a DFA to accept $L_{\ell,n,\mathcal{R}}$, and not less than $\sum_{i=1}^{s} p_i^{\alpha_i}$ states on two-way DFAs and NFAs, and on isolated cut-point PFAs are needed.

Acknowledgements. The authors wish to thank the anonymous referees.

References

1. Aho, A.V., Lam, M.S-L., Sethi, R., Ullman, J.D.: Compilers: principles, techniques, and tools, 2 Ed., Addison-Wesley (2006)
2. Bednárová, Z., Geffert, V., Mereghetti, C., Palano, B.: Removing nondeterminism in constant height pushdown automata. In: DCFS 2012. LNCS **7386**, pp. 76–88. Springer (2012)
3. Bednárová, Z., Geffert, V., Mereghetti, C., Palano, B.: Boolean language operations on nondeterministic automata with a pushdown of constant height. In: CSR 2013. LNCS **7913**, pp. 100–111. Springer (2013)
4. Bednárová, Z., Geffert, V., Mereghetti, C., Palano, B.: Removing nondeterminism in constant height pushdown automata. Inf. Comp. **237**, 257–267 (2014)
5. Bednárová, Z., Geffert, V., Mereghetti, C., Palano, B.: Boolean language operations on nondeterministic automata with a pushdown of constant height. J. Comp. Sys. Sci. **90**, 99–114 (2017)
6. Bertoni, A., Mereghetti, C., Palano, B.: Trace monoids with idempotent generators and measure-only quantum automata. Nat. Comp. **9**, 383–395 (2010)
7. Bianchi, M.P., Mereghetti, C., Palano, B.: On the power of one-way automata with quantum and classical states. In: CIAA 2014. LNCS **8587**, pp. 84–97. Springer (2014)
8. Bianchi, M.P., Mereghetti, C., Palano, B.: Size lower bounds for quantum automata. Th. Comp. Sci. **551**, 102–115 (2014)

9. Bianchi, M.P., Mereghetti, C., Palano, B.: Quantum finite automata: Advances on Bertoni's ideas. Th. Comp. Sci. **664**, 39–53 (2017)
10. Bianchi, M.P., Mereghetti, C., Palano, B., Pighizzini, G.: On the size of unary probabilistic and nondeterministic automata. Fund. Inf. **112**, 119–135 (2011)
11. Bordihn, H., Fernau, H., Holzer, M., Manca, V., Martín-Vide, C.: Iterated sequential transducers as language generating devices. Th. Comp. Sci. **369**, 67–81 (2006)
12. Choffrut, C., Malcher, A., Mereghetti, C., Palano, B.: On the expressive power of FO[+]. In: LATA 2010. LNCS **6031**, pp. 190–201. Springer (2010)
13. Choffrut, C., Malcher, A., Mereghetti, C., Palano, B.: First-order logics: some characterizations and closure properties. Acta Inf. **49**, 225–248 (2012)
14. Chrobak, M.: Finite automata and unary languages. Th. Comp. Sci. **47**, 149–158 (1986) – Corrigendum, *ibid* **302**, 497–498, (2003)
15. Citrini, C., Crespi-Reghizzi, S., Mandrioli, D.: On deterministic multi-pass analysis. SIAM J. Comput. **15**, 668–693 (1986)
16. Feletti, C., Mereghetti, C., Palano, B.: Uniform circle formation for swarms of opaque robots with lights. In: SSS 2018. LNCS **11201**, pp. 317–332. Springer (2018)
17. Friburger, N., Maurel, D.: Finite-state transducer cascades to extract named entities in texts. Th. Comp. Sci. **313**, 93–104 (2004)
18. Ginzburg, A.: Algebraic theory of automata. Academic Press (1968)
19. Jakobi, S., Meckel, K., Mereghetti, C., Palano, B.: Queue automata of constant length. In: DCFS 2013. LNCS **8031**, pp. 124–135. Springer (2013)
20. Kutrib, M., Malcher, A., Mereghetti, C., Palano, B.: Descriptional complexity of iterated uniform finite-state transducers. In: DCFS 2019. LNCS **1612**, pp. 223–234. Springer (2019)
21. Kutrib, M., Malcher, A., Mereghetti, C., Palano, B.: Iterated uniform finite-state transducers. In: ICTCS 2019. CEUR WORKSHOP PROCEEDINGS, vol. 2504, pp. 52–57. CEUR-WS.org (2019)
22. Kutrib, M., Malcher, A., Mereghetti, C., Palano, B.: Iterated uniform finite-state transducers: Descriptional complexity of nondeterminism and two-way motion. In: DCFS 2020. LNCS **12442**, pp. 117–129. Springer (2020)
23. Kutrib, M., Malcher, A., Mereghetti, C., Palano, B.: Deterministic and nondeterministic iterated uniform finite-state transducers: Computational and descriptional power. In: CiE 2020. LNCS **12098**, pp. 87–99. Springer (2020)
24. Kutrib, M., Malcher, A., Mereghetti, C., Palano, B.: Iterated uniform finite-state transducers on unary languages In: SOFSEM 2021. LNCS **12607**, pp. 218–232. Springer (2021)
25. Kutrib, M., Malcher, A., Mereghetti, C., Palano, B.: Descriptional complexity of iterated uniform finite-state transducers. Inf. Comp. (In press)
26. Manca, V.: On the generative power of iterated transductions. In: Words, Semigroups, and Transductions. pp. 315–327. World Scientific (2001)
27. Mereghetti, C.: Testing the descriptional power of small Turing machines on nonregular language acceptance. Int. J. Found. Comput. Sci. **19**, 827–843 (2008)
28. Mereghetti, C., Palano, B.: The complexity of minimum difference cover. J. Discr. Alg. **4**, 239–254 (2006)
29. Mereghetti, C., Palano, B., Pighizzini, G.: Note on the succinctness of deterministic, nondeterministic, probabilistic and quantum finite automata. Th. Inf. Appl. **35**, 477–490 (2001)
30. Mereghetti, C., Pighizzini, G.: Two–way automata simulations and unary languages. J. Autom., Lang. Comb. **5**, 287–300 (2000)
31. Mereghetti, C., Pighizzini, G.: Optimal simulations between unary automata. SIAM J. Comput. **30**, 1976–1992 (2001)